CLAIMS

1.   A processor switchable between a first
execution mode and a second execution mode, the
processor having a first processor context when in the
first execution mode and a second processor context,
different from the first processor context, when in the
second execution mode, wherein the processor is
arranged to generate an exception when the processor
attempts to change from one execution mode to the
other.

2.   A processor according to claim 1 wherein the
second processor context is larger than the first
processor context and the exception is generated when
the processor attempts to change from the first
execution mode to the second execution mode.

3.   A processor according to claim 2, wherein the
processor is arranged to preserve the second processor
context, or that part of the second processor context
which is different from the first processor context,
when said exception has been generated.

4.   A processor according to claim 2, wherein the
processor is arranged to execute a plurality of threads
on a time share basis, and the processor is arranged
such that, when the processor is switched to a thread
which is in the first execution mode, or when the
processor is switched to a thread which was the last
thread to be in the second execution mode, only the
first processor context is preserved.

5.   A processor according to claim 4, wherein the
second processor context, or that part of the second
processor context which is different from the first

processor context, is preserved when the processor next
enters the second execution mode to execute a thread
other than the last thread to be in the second
execution mode.

6.   A processor according to claim 2, wherein the
processor is arranged to execute a plurality of threads
on a time share basis, and the number of threads that
may be in the second execution mode at any one time is
less than the total number of threads that may be
active on the processor at any one time.

7.   A processor according to claim 6, wherein the
processor is arranged such that, when said exception
has been generated, a check is carried out to determine
whether the thread that caused the exception is allowed
to enter the second execution mode.

8.   A processor according to claim 7, wherein the
check comprises determining whether that thread is a
thread which is barred from the second execution mode.

9.   A processor according to claim 7, wherein the
check comprises determining whether a predetermined
number of other threads are already in the second
execution mode.

10.   A processor according to claim 9, wherein the
processor is arranged such that, if a predetermined
number of other threads are already in the second
execution mode, execution of the thread that caused the
exception is suspended until the number of other
threads that are in the second execution mode is less
than the predetermined number.

11.   A processor according to claim 1, wherein the

processor is arranged to execute a first instruction
set when in the first execution mode and a second
instruction set when in the second execution mode.

5   12.   A processor according to claim 1, wherein the
processor is switchable between a supervisor mode and a
user mode, the user mode having restricted access to
the processor's resources in comparison to the
supervisor mode, and, when said exception is generated,
10   the processor transfers from the user mode to the
supervisor mode.

      13.   A processor according to claim 1, the
processor comprising at least one execution unit and a
15   plurality of storage locations, the first processor
context comprising the contents of storage locations
accessible in the first execution mode and the second
processor context comprising the contents of storage
locations accessible in the second execution mode.

20   14.   A processor according to claim 1, the
processor comprising a plurality of computational units
for executing instructions in parallel, each
computational unit having at least one execution unit
25   and at least one storage location to which the
execution unit has access.

      15.   A processor according to claim 1, wherein the
first execution mode is a scalar mode and the second
30   execution mode is a parallel mode.

      16.   A processor switchable between a first
execution mode and a second execution mode, the
processor having a first processor context when in the
35   first execution mode and a second processor context,
larger than the first processor context, when in the

second execution mode, wherein the processor is arranged to execute a plurality of threads on a time share basis, and the processor is arranged such that, when the processor switches to a thread which is in the first execution mode, or when the processor switches to a thread which was the last thread to be in the second execution mode, only the first processor context is preserved.

17. A processor according to claim 16, wherein the second processor context, or that part of the second processor context which is different from the first processor context, is preserved when the processor next enters the second execution mode to execute another thread.

18. A processor according to claim 16, wherein the processor is arranged such that the number of threads that may be in the second execution mode at any one time is less than the total number of threads that may be active on the processor at any one time.

19. A processor according to claim 16, wherein the first execution mode is a scalar mode and the second execution mode is a parallel mode.

20. A processor switchable between a first execution mode and a second execution mode, the processor having a first processor context when in the first execution mode and a second processor context, larger than the first processor context, when in the second execution mode, wherein the processor is arranged to execute a plurality of threads on a time share basis, and the processor is arranged such that the number of threads that may be in the second execution mode at any one time is less than the total

number of threads that may be active on the processor
at any one time.

21.   A processor according to claim 20 wherein the
processor is arranged such that, when a thread attempts
to enter the second execution mode, a check is carried
out to determine whether that thread is allowed to
enter the second execution mode.

22.   A processor according to claim 21, wherein
the check comprises determining whether that thread is
a thread which is barred from the second execution
mode.

23.   A processor according to claim 21, wherein
the check comprises determining whether a predetermined
number of other threads are already in the second
execution mode.

24.   A processor according to claim 23, wherein
the processor is arranged such that, if a predetermined
number of other threads are already in the second
execution mode, the thread that attempted to enter the
second execution mode is prevented from entering the
second execution mode until the number of other threads
that are in the second execution mode is less than the
predetermined number.

25.   A method of operating a processor, the
processor being switchable between a first execution
mode and a second execution mode and having a first
processor context when in the first execution mode and
a second processor context, different from the first
processor context, when in the second execution mode,
the method comprising generating an exception when the
processor attempts to change from one execution mode to

the other.

26. A method of operating a processor, the processor being switchable between a first execution mode and a second execution mode and having a first processor context when in the first execution mode and a second processor context, larger than the first processor context, when in the second execution mode, the method comprising executing a plurality of threads on a time share basis, and preserving only the first processor context when execution switches to a thread which is in the first execution mode, or when execution switches to a thread which was the last thread to be in the second execution mode.

27. A method of operating a processor, the processor being switchable between a first execution mode and a second execution mode and having a first processor context when in the first execution mode and a second processor context, larger than the first processor context, when in the second execution mode, the method comprising executing a plurality of threads on a time share basis and limiting the number of threads that may be in the second execution mode at any one time to less than the total number of threads that may be active on the processor at any one time.

28. A computer readable storage medium having stored thereon an operating system for a processor which is switchable between a first execution mode and a second execution mode and which has a first processor context when in the first execution mode and a second processor context, different from the first processor context, when in the second execution mode, the operating system comprising an exception handling program portion for handling an exception generated

when the processor attempts to change from one
execution mode to the other.

     29. A computer readable storage medium having
stored thereon an operating system for a processor
which is switchable between a first execution mode and
a second execution mode and which has a first processor
context when in the first execution mode and a second
processor context, larger than the first processor
context, when in the second execution mode, the
operating system comprising a program portion for
switching execution between a plurality of threads on a
time share basis, and a program portion for preserving
only the first processor context when execution
switches to a thread which is in the first execution
mode, or when execution switches to a thread which was
the last thread to be in the second execution mode.

     30. A computer readable storage medium having
stored thereon an operating system for a processor
which is switchable between a first execution mode and
a second execution mode and which has a first processor
context when in the first execution mode and a second
processor context, larger than the first processor
context, when in the second execution mode, the
operating system comprising a program portion for
switching execution between a plurality of threads on a
time share basis, and a program portion for limiting
the number of threads that may be in the second
execution mode at any one time to less than the total
number of threads that may be active on the processor
at any one time.